Anonymous Referee #1

We would like to thank the reviewer for helpful comments and suggestions. Please find our responses to individual comments below in bold. The corresponding changes to the manuscript are highlighted with orange text in the supplement.

General comments Its a good idea to provide a library that will make netcdf IO in fortran easier, even at the cost of performance (at least for prototyping). The paper is fairly easy to read. It was very useful to be able to read the fortran example provided on github (test_ncio.f90), so it would probably be a good idea mention the github repository at the beginning of the paper and also include a "Download" section on github page to help novice users download the code eg git clone `https://github.com/alex-robinson/ncio.git` **We have added information about the github repository in the abstract, as well as separated the section in the Discussion concerning how to obtain the code. There is a direct download link available on the Github repository, so we have not added additional information on that page.**

Other comments

line 25: the paper should mention the CMOR (Climate Model Output Rewriter) library `http://www2-pcmdi.llnl.gov/cmor` and the CF (Climate and Forecast) conventions `http://cfconventions.org/` Explicitly mentioning CF will also make it clearer for some people why you are using the long_name/standard_name/... attributes later. Are you speaking about CF on page 304 when you mention "convention attributes"? **We have added a specific reference to CMOR and CF. conventions attributes was a typo, this has been changed just to typical attributes.**

line 23: there should be some more explanation about why and when some people have to use nc_write_map. Is it something you need when using dimensions defined in km? **Additional text has been added. Indeed the map definition is related to using a regional projection as opposed to a latlon grid.**

There should be a note about the use of the extended dimension and how, for instance, the time axis can be extended with repeated calls to nc_write. test_ncio.f90 uses the unlimited=TRUE parameter in nc_write_dim, but the documentation of nc_write_dim on page 315 does NOT mention this useful parameter **This has been clarified in the revised manuscript, thank you for pointing this out.**

The paper does not mention if the library will work with netcdf3 or

netcdf4 **We have added an argument to the nc_create function to allow the user to choose between writing in the netcdf3 or netcdf4 format, as well as to specify whether previous files should be overwritten.**

The paper does not mention at all how the errors are handled. **We have added a sentence to the revised text. Basically ncio outputs the native NetCDF error and stops the program.**

What happens if you try to nc_create on an existing file? Do you get an error or will the existing file be erased? **The existing file will be erased by default. We have added an argument to nc_create to allow the user to modify this behavior.**

It is a limitation (an acceptable limitation) to be able to only read and write string attributes. It could be a good idea to use a more explicit name for the functions nc_read_attr =¿ nc_read_str_attr nc_write_attr =¿ nc_write_str_attr **This function has been modified it is now possible to write and read attributes as strings or numeric values.**

The tables describing the function calls mix input and output parameters and do not explicitely list the allowed types. This should be improved e.g, p 311, nc_read_attr value (OUT) value of the STRING attribute to be read **This is a good suggestion, and we have added a clear OUTPUT: phrase where appropriate.**

Typos ===== page 302, line 4, cUmbersome, "module NCIO" -¿ "NCIO module" **Done.**

line 11, revolutionized? **This seems clear to us.**

line 23, common TASKS? **Done.**

page 303, line 5, NCIO module **Done.**

line 9, hopes-¿hope? **Done.**

line 20, analOgous? **Done.**

line 21, "the subroutine will read all the variable"? **We could not find what this comment refers to.**

page 304, line 13, "like with"? **Changed to as with.**

page 305, line 12, "be prepared" =¿ "be ready"? **Done.**

line 17, exception **Done.**

page 306, line 15, to A file **Done.**

line 27, use the correct "ff" in "affect" **This comment is not clear to us.**

page 307 and 308, it would be nice to rewrite a bit the "Conclusion" (revert a bit the order of some sentences?) **On this point, we have to disagree, although it is arguably a small concern.**

p 310, specify in another way that nc_size returns an integer. **We have added the OUTPUT: text here as well.**

p 313, varname, "name of the variable the attribute should be associated to" **Done.**

p 315, add the "unlimited" parameter **Done.**

p 316, remove "fortran data type of" for the dat parameter **Done.**

Anonymous Referee #2

The new tool NCIO presented in this paper is a binding between Fortran and some major NetCDF functions. As Fortran is a widely used programming language in modelisation, I think the NCIO tool could be very useful. This wrapper library is a bridge between those two tools and it gives Fortran users access to a new set of tools without technical investment. A key design point I would like to highlight: NCIO is hiding unecessary steps where the user has to create some intermediate variables while using netCDF functions. This is good design to take those complications away from the user. The source code which is hosted by GitHub is more than handy, it is also great for collaboration. Anyone can help the project: -adding features -correct unexpected behaviours -discuss issues or bugs and everything is happening in public for the world to see. The "all in one file" keeps the using simple for the user. Maybe two versions could exist: one for production like the current state of the project and the other one where the unique .f90 file is splitted in several files for more modularity, but more importantly to be more developper-friendly if some day a volunteer want to help the NCIO project. README.md is clear for anyone to use NCIO.

We thank the second reviewer for the positive comments, and we will keep the possibility of separating the module into several files in mind for the future. To this end we have opened an issue on the github repository suggesting this as a potential development direction.